Name : **Arjun Paudel**

Student ID Number : **S2271954**

Subject : **Software Development For Data Science**

Module Code : **MMI226822**

Coursework 2

Submission Date : **15 Janurary 2024**

Customer Churn Prediction By Using Machine Learning Classification Algorithm

Abstract

This paper show how machine learning algorithm can be used to predict the churn behaviour of customer in any organization. First the data (The Telco Customer Churn Data) was taken from the kaggle website which provide the information of customer that purchased home phone and internert services who stay or leave the company. In first phase data was clean using different techinque (cleaning null values, cleaning duplicate values, find out the outliers). Then machine learning model that suit this data was discussed and with the help of encoding, the full data is made ready for machine leaning task. Then Decision tree classifer model is introduced and with the help of confusion metrix the evualtion of model was done.

Introduction

The main objective of this analysis is to predcit the value of target variable by using the different features columns in machine learning. And the machine learning task used in this analysis is Classification. Classification algorithms are a type of supervised learning algorithm that are designed to categorize instances into predefined classes. These algorithm are generally used to make prediction on unseen data. Decision tree split the dataset based on significant features in tree like structure. Strength of Decision tree classifer is interpretable, handles non-linear relationship between target and features columns.

Dataset that was choosen on this analysis is Telco Customer Churn Data (IBM), which provides the churn behaviour of different customer who has taken the home phone and internet services. Churn is a crucial metrix for some business, so to predict the this(churn) behaviour of customer, classification machine learning model is used so that company or

organization that provides these service may improved their facilities so that the customer will not leave them after certain time.

Dataset Selection

Customer Churn Predictions

Dataset:Telco Customer Churn

Telco Customer Churn data (IBM) is a dataset which is taken from the kaggle website (https://www.kaggle.com/datasets/blastchar/telco-customer-churn/data). This dataset provide the information about the fictional telco company which provide phone and internet services towards customers in California USA.

This dataset has an information of 7043 customer whether the customer has left, stayed or signed up for the services distributed over 21 different variables.This dataset contain four main infomation and that is demeographic information of customers (age, if they have dependent and partners, gender), account informatio of customer (Customer ID, monthly charge, total charge, how long they continue with the services, billing information,contract type), service information (such as phone services, online security, online banking, device protection, internert serices) and the last one is information about churn that is the customer who left on last month.

This dataset was chosen due to its richness and applicability to the aim of predicitng the loss of customers. Also this dataset provides comprehensive understanding of the variables impacting churn by capturing a variety of customer interactions. Furthermore, the avialabilty of both categorical and numerical variables provides the more information for developing machine learning model.

Data Ingestion/ Cleaning / Wrangling / EDA

Importing Necessary Libraries

First all the libraries that are needed through out the analysis were imported,

In [1]:
```
#Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
```

Data Ingestion

First, accessing google drive from google colab to load the dataset into dataframe

In [2]:
```
#acessing google drive from google colab
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [3]:
```
#reading data from drive and load it into dataframe
df = pd.read_csv('/content/drive/MyDrive/SDFDS/cw2/Telco-Customer-Churn.csv')
```

Data Information

Shape of data

In [4]:
```
df.shape
```

Out[4]:
```
(7043, 21)
```

It shows that the total number of obeseavation of this data is about 7043 where the total number of column variable is 21.

Data Type

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

The dataset is combine with the data type of integer, float and object.

Here is the details descriptions of all 21 variables

1. **customerID**: A unique number that was given to each customer.
2. **gender**: Gender of customer that is Male or Female.
3. **SeniorCitizen**: Show if customer is sixty-five or older that is 1 or 0.
4. **Partner**: Showing if customer is married that is Yes or No.
5. **Dependents**:Showing if customer live with any dependent such as children, parent and represented by Yes or No
6. **tenure**: How long does customer stay with company, represented in the total number of month
7. **PhoneService**: Show if customer used phone services, represented in Yes or No.
8. **MultipleLines**: Show if customer used multiple phone services, represented in Yes, No and No phone serivces.
9. **InternetService**:Show if customer used internet services, represented in No, Fiber optic, DSL.
10. **OnlineSecurity**:Show if customer deposit online security, represented in Yes, No and No internet serivces.
11. **OnlineBackup**:Show if customer used online backup services, represented in Yes, No and No internrt serivces.
12. **DeviceProtection**:Show if customer used device protection services, represented in Yes, No and No internrt serivces.
13. **TechSupport**: Show if customer used tech support services, represented in Yes, No and No internrt serivces.
14. **StreamingTV**: Show if customer used streaming TV services, represented in Yes, No and No internrt serivces.
15. **StreamingMovies**:Show if customer used streaming movies services, represented in Yes, No and No internrt serivces.
16. **Contract**: Show customer contract type, represented in month-to-month, one year and two year.
17. **PaperlessBilling**:Show customer paperless billing type, represented in Yes, No.
18. **PaymentMethod**: Showing cusomer payment menthod that is Bank transfer(automatic), Credit card (automatic), Electronic check and Mailed check
19. **MonthlyCharges**: Showing total monthly charger of the all services.
20. **TotalCharges**: Shoing the total charge of the customer during staying period.
21. **Churn**: Show the information whether customer end the relationship with company or not, represented in boolean form Yes or No.

Data Variable

Showing which one is categorical variable and which one is numerical variable

In [6]:
```
#finding numerical and categorical values
numerical_variable = df.select_dtypes(include = [np.number])
categorical_variable = df.select_dtypes(include = [object])
```

In [7]:
```
#seeing numerical column
numerical_variable.columns
```

Out[7]:
```
Index(['SeniorCitizen', 'tenure', 'MonthlyCharges'], dtype='object')
```

In [8]:
```
#seeing categorical column
categorical_variable.columns
```

Out[8]:
```
Index(['customerID', 'gender', 'Partner', 'Dependents', 'PhoneService',
       'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup',
       'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
       'Contract', 'PaperlessBilling', 'PaymentMethod', 'TotalCharges',
       'Churn'],
      dtype='object')
```

Showing Few First data from the dataframe

In [9]:
```
df.head(5)
```

Out[9]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service
1	5575-GNVDE	Male	0	No	No	34	Yes	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service
4	9237-HQITU	Female	0	No	No	2	Yes	No

5 rows × 21 columns

Summary Statistics of Numerical values

In [10]:
```
df.describe()
```

Out[10]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

Data Cleaning

The most important part of the data analaysis is to cleaning the data.

Finding Missing Values

In [11]:
```
#checking missing value
df.isnull().sum()
```

Out[11]:
```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

The result shows that there is no any missing values present in the data.

Checking Duplicate Data

In [12]:
```
#checking the duplicate data
df.loc[df.duplicated()]
```

Out[12]:

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	I

0 rows × 21 columns

The results show that there is no any duplicate data in the dataframe.

Checking Outliers

Let see if there is any outliers in the data.

For that first **tenure** column was checked if there is any value zero which means the total duration of customer having phone and internet service is zero.

In [13]:
```
zero_tenure_row = df[df['tenure'] == 0]
zero_tenure_row
```

Out[13]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLin
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phor servi
753	3115-CZMZD	Male	0	No	Yes	0	Yes	N
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	N
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Y
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phor servi
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	N
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Y
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	N
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	N
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Y
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Y

11 rows × 21 columns

Here, it shows that there are 11 such data which total duration zero and also in TotalCharge column there is no any data, but when we check for null values before it doesn't show at that time. This show that it is clearly outlier which automatically affect our model while predicting the churn behaviour of customer. So to overcome this default, all the rows with tenure having zero will get drop from the dataframe.

In [14]:
```
df = df.drop(zero_tenure_row.index)
```

Now these rows are drop from the dataset.

Here is the new size of the dataframe.

In [15]:
```
df.shape
```

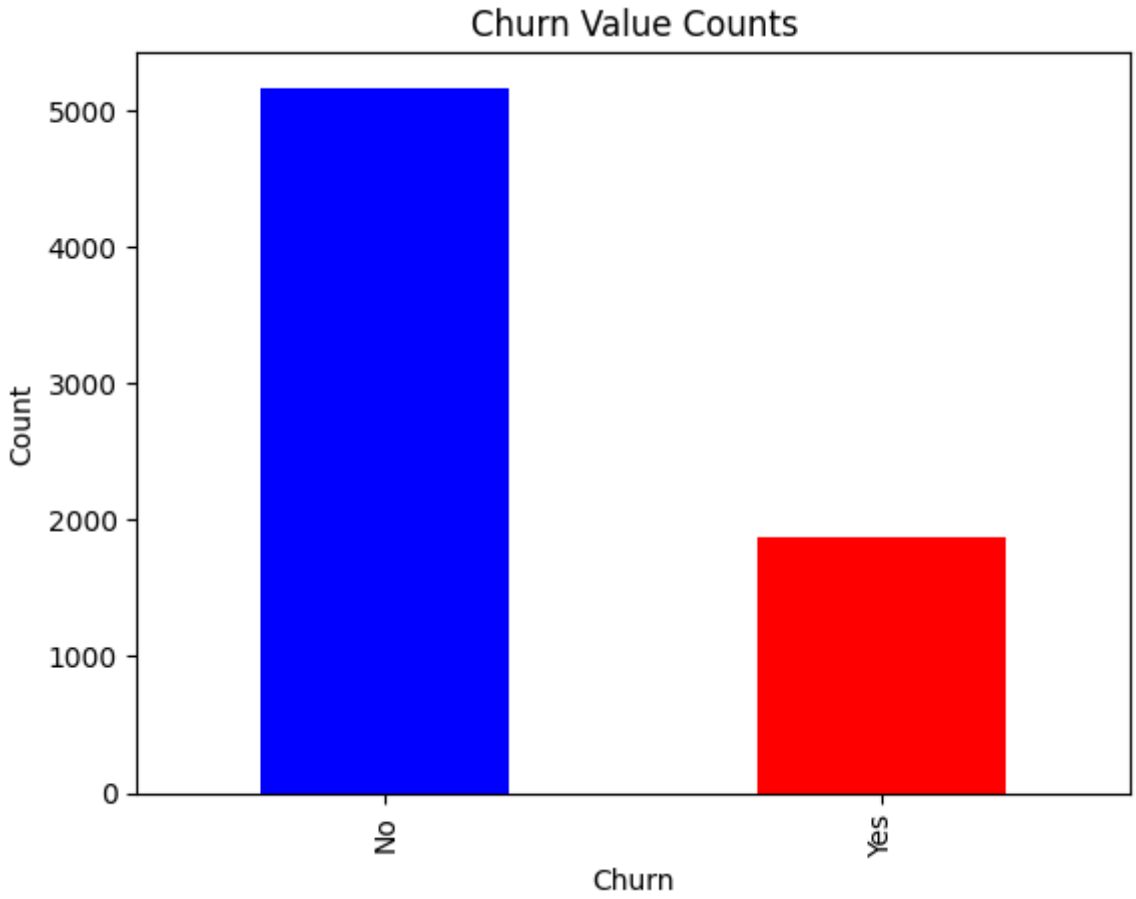
Out[15]:
```
(7032, 21)
```

Exploratory Data Analysis

Doing exploratory data analysis to find out which machine learning algorithm should apply to target variable

In [16]:
```
#counting total number of unique churn values
churn_values = df['Churn'].value_counts()

# Plot using a bar plot
churn_values.plot(kind='bar', color=['blue', 'red'])
plt.title('Churn Value Counts')
plt.xlabel('Churn')
plt.ylabel('Count')
plt.show()
```



From this analysis it show that there are only two values in whole Churn column that is Yes and No. Also this show that majority of customer extend there contract.

Task Defination / Formulating Machine Learning Problems

Machine Learning Task: Classification

Here classficiation algorithm specially Decision Tree Classifier algorthim was used to predeict the customer churn. Main objective of this analysis is to find whether customer is like to churn or not, and the **target variable Churn** has a boolean values that 'Yes',or 'No' which was found by the exploratory data analaysis.

Classification Algorithm

Classification algorithms are a type of supervised learning algorithm that are designed to categorize instances into predefined classes. These algorithm are generally used to make prediction on unseen data. Decision tree split the dataset based on significant features in

tree like structure. Strength of Decision tree classifer is interpretable, handles non-linear relationship between target and features columns.

Data Encoding

Before creating model, data should be transform that is suitable for classification, which is in binary form. So from the dataframe binarise few column (such as geneder,Dependents, phoneservice, internetservice, MultipleLines,InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, PaperlessBilling) data. So for that these data was changed to binary form, **'Yes': 1, 'No': 0, 'No phone service': 0, 'No internet service':0 , 'Female':0, 'Male':1**

In [17]:
```
#binarise the data that is suitable for classification
df = df.replace({'Yes': 1, 'No': 0,'No phone service': 0,'No internet service':0 ,'
```

If the data was checked, then it looks like this.

In [18]:
```
df.head()
```

Out[18]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	7590-VHVEG	0	0	1	0	1	0	0
1	5575-GNVDE	1	0	0	0	34	1	0
2	3668-QPYBK	1	0	0	0	2	1	0
3	7795-CFOCW	1	0	0	0	45	0	0
4	9237-HQITU	0	0	0	0	2	1	0

5 rows × 21 columns

Still in the dataset, there are few string data. If the classifiaction algorthim is run through these data, it will not executes. That's why here we introduce the **one hot encoding**. One hot encoding creates a uniques column for each values for each variables. SO here in the three variables(InternetService ,Contract, PaymentMethod) one hot encoding was introduce.

In [19]:
```
# Pandas approach to one hot encoding
features_to_onehot_encode = ["InternetService","Contract","PaymentMethod"]
for f in features_to_onehot_encode:
    df = pd.concat([df, pd.get_dummies(df[f], prefix=f)], axis=1)
    df = df.drop(columns=[f])
```

Let see how the dataframe looks like now.

In [20]:
```
df.columns
```

Out[20]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'PaperlessBilling', 'MonthlyCharges', 'TotalCharges',
       'Churn', 'InternetService_0', 'InternetService_DSL',
       'InternetService_Fiber optic', 'Contract_Month-to-month',
       'Contract_One year', 'Contract_Two year',
       'PaymentMethod_Bank transfer (automatic)',
       'PaymentMethod_Credit card (automatic)',
       'PaymentMethod_Electronic check', 'PaymentMethod_Mailed check'],
      dtype='object')
```

As one hot encoding was done, there is seperate column name based on these three variable with their uniques values. But one things to notice over here "InternetService_0" column show information about the no internet services, and there are another two columns 'InternetService_DSL', 'InternetService_Fiber optic', shows that the customer has the internet services of DSL, and Fiber Optics. So the "InternetService_0" is not necessary while predicting the target variables so this column is droped in next step.

In [21]:
```
#dropping
df = df.drop('InternetService_0', axis=1,)
```

Also the column name of the variable are not in standard order, some variable name has space on it, so to changed it into standard format following code was run.

In [22]:
```
#replacing space in column variable name to "_".
new_columns = [col.replace(' ', '_') for col in df.columns]
df.columns = new_columns
```

Finally data is ready to create a model.

Creating A Model

Here Decision Tree Classifier Model is used to predict the target values. For that first the target variable was loaded into the y and predicting feature variable was loaded into the X.

In [23]:
```
#target variable
y = df[['Churn']].copy()
y
```

Out[23]:

	Churn
0	0
1	0
2	1
3	0
4	1
...	...
7038	0
7039	0
7040	0
7041	1
7042	0

7032 rows × 1 columns

First all the predicting feature was gathered as a list where, two columns were not selected, first one is 'Churn' which is also a target variables, and another one is 'customerID', which has no any relationship with the predicting the model.

In [24]:
```
#predicing feature columns
predicting_data = ['gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'PaperlessBilling', 'MonthlyCharges', 'TotalCharges',
       'InternetService_DSL', 'InternetService_Fiber_optic',
       'Contract_Month-to-month', 'Contract_One_year', 'Contract_Two_year',
       'PaymentMethod_Bank_transfer_(automatic)',
       'PaymentMethod_Credit_card_(automatic)',
       'PaymentMethod_Electronic_check', 'PaymentMethod_Mailed_check' ]
```

In [25]:
```
#features variables
X = df[predicting_data].copy()
```

Here data is split into training and testing data with the help of sklearn libraries. Here 80% data is split into training phase where as remaining 20% data is used for testing purpose, and the random_stage function helps to seed the random number generator for reproducibility which is 324.

In [26]:
```
# split data in train and test set
# X = input data, y = output labels
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

Here the decision tree classifier model is used to create a model which has a parameter of max leaf node 10, which means the tree will stop after reaching leaf 10.

In [27]:
```
# fit the model on train set
churn_prediction = DecisionTreeClassifier(max_leaf_nodes=10, random_state=0)
churn_prediction.fit(X_train, y_train)
```

Out[27]:
```
DecisionTreeClassifier(max_leaf_nodes=10, random_state=0)
```

Predicitng the model

Now with the help of model, we will predict the values on testing data of X variables.

In [28]:
```
#predicting the data on X testing data
predict = churn_prediction.predict(X_test)
predict[:10]
```

Out[28]:
```
array([1, 0, 0, 1, 0, 1, 0, 0, 0, 0])
```

In [29]:
```
y_test['Churn'][:10]
```

Out[29]:
```
371     1
871     0
1235    0
4526    0
5105    1
248     1
2545    0
1315    0
3502    0
6583    0
Name: Churn, dtype: int64
```

Evaluation Metrics

In [30]:
```
#showing model accuracy
accuracy_score(y_true = y_test, y_pred = predict)
```

Out[30]:
```
0.7903340440653873
```

This shows that 79.03% of prediction made by the model is correct.

In [31]:
```
report = classification_report(y_test, predict)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.83      0.91      0.87      1043
           1       0.63      0.46      0.53       364

    accuracy                           0.79      1407
   macro avg       0.73      0.68      0.70      1407
weighted avg       0.78      0.79      0.78      1407
```

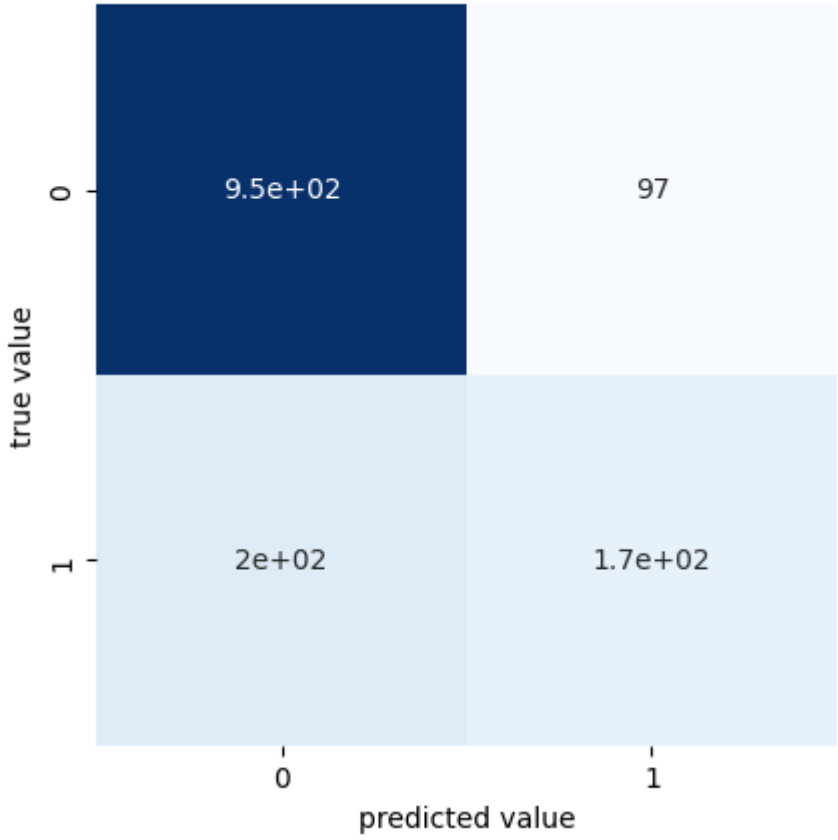
This classification reports show the result of overall performace of the model on predicting the correct values.

1. Precision: This show the accuracy of the positive prediction by the modesl
2. Recall: It shows that how model is able to capture all the relevant instance of class from the dataframe.
3. F1-Score: It show the harmonic mean of precision and recall.

4. Support: Show total number of occurrence of specifi class in dataset.

To know it in better, let's plot a confusion maxtrix, Confusion matrix helps to provide indepth insight from the prediction. Also if there is any error on the model, can be easily show through confusion matrix.

In [33]:
```
# Creating a confusion matrix
matrix = confusion_matrix(y_true=y_test, y_pred=predict)
sns.heatmap(matrix, square=True, annot=True, cbar=False, cmap="Blues")
plt.xlabel('predicted value')
plt.ylabel('true value');
```



From this confusion matrix, the machine leaning analysi can be conclude as the, model has made error on total of 297 sample. On which 97 are false postivie and 200 (2e+02) are false negative error.

Result, Discussion and Conclusion

Result

The overall accuracy of the model is 0.7903340440653873 and the total number of error made on predicting the sample data is 297 which is found with the help of confusion metrix.

Form the result of exploratory data analysis of target variable (Churn), it show that there is high number of customer stay with organization.

Challenges

Due to the imbalanced in dataset, it hard to predict the actual result and another challenging part is due to mixed datatype in same field data has been represetned in string

format and numeric format. Another one is in total charge column there is a data in string format where they just put the values by entering space which makes difficulties in finding the null values.

Limitation

There is some limitation while collecting the data, unwanted data were also in the dataset such as in tenure column, customer who hasn't stayed with the organization that is tenure is 0, was also in collected in dataset, which directly impact the total charge field.

Possible Improvement for furture

Try to collect the diverse data so that model doesn't get baised towards one sided. Exploring advanced algorithms could also enhance the model performance.

References

Kotsiantis, S.B., Zaharakis, I.D. and Pintelas, P.E., 2006. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26, pp.159-190

Huang, B., Kechadi, M.T. and Buckley, B., 2012. Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1), pp.1414-1425.

Dahiya, K. and Bhatia, S., 2015, September. Customer churn analysis in telecom industry. In 2015 *4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)* (pp. 1-6). IEEE.

Lu, J., 2002. Predicting customer churn in the telecommunications industry—An application of survival analysis modeling using *SAS. SAS User Group International (SUGI27) Online Proceedings*, 114, p.27.